

*"Optimizing the performance of computer systems has always been an art relegated to a few individuals who happen to have the 'right skills'."*

**Amir H. Majidimehr**

Optimizing Unix for Performance, 1995



## Методика нагрузочного тестирования

# Оглавление

|      |  |    |
|------|--|----|
| 1    | История внесения изменений в документ                | 4  |
| 2    | Лист согласования                                    | 5  |
| 3    | Список терминов и сокращений                         | 6  |
| 4    | Введение   | 7  |
| 4.1  | Назначение документа                                 | 7  |
| 4.2  | Объект тестирования                                  | 7  |
| 5    | Цели и задачи  | 9  |
| 6    | Ограничения тестирования                             | 11 |
| 7    | Архитектура системы                                  | 12 |
| 7.1  | Конфигурация серверов продуктивного стенда           | 12 |
| 8    | Взаимодействие с внешними системами                  | 15 |
| 9    | Стратегия тестирования                               | 17 |
| 9.1  | Этапы тестирования                                   | 17 |
| 10   | Моделирование нагрузки                               | 18 |
| 11   | Тестовый стенд                                       | 20 |
| 11.1 | Архитектура тестового стенда                         | 20 |
| 11.2 | Конфигурация тестового стенда                        | 20 |
| 11.3 | Конфигурация ПО                                      | 21 |
| 11.4 | Тестовые данные для средств НТ                       | 21 |
| 11.5 | Методика удаления тестовых данных после тестирования | 21 |
| 11.6 | Прочие требования, предъявляемые к тестовой среде    | 21 |
| 12   | Анализ статистики                                    | 22 |
| 12.1 | Наиболее часто используемые операции                 | 22 |
| 13   | Профили нагрузки                                     | 24 |
| 13.1 | Процессинговая деятельность (МК)                     | 24 |
| 14   | Сценарии пользования                                 | 25 |
| 15   | Наполнение БД  | 27 |
| 16   | Планируемые тесты                                    | 29 |
| 16.1 | Перечень типов тестов                                | 29 |
| 16.2 | Планируемые тесты                                    | 29 |
| 16.3 | Критерии успешности проведенного теста               | 30 |
| 17   | Требования к производительности                      | 31 |

|      |  |    |
|------|--|----|
| 18   | Мониторинг производительности                      | 32 |
| 18.1 | Метрики производительности                         | 32 |
| 18.2 | Способы мониторинга показателей производительности | 33 |
| 19   | Риски проекта                                      | 35 |
| 20   | Требования к заказчику                             | 36 |
| 21   | Материалы, подлежащие сдаче                        | 38 |
| 22   | Оценка точности проведения нт                      | 39 |
| 23   | Приложения   | 43 |
| 24   | Контакты   | 44 |

# 1 История внесения изменений в документ

| Дата       | Версия | Описание        | Автор          |
|------------|--------|-----------------|----------------|
| 09.09.2024 | 0.1    | Документ создан | Оберган Никита |
|            |        |                 |                |
|            |        |                 |                |

## 2 Лист согласования

Заполняется согласующими лицами со стороны заказчика.

| ФИО                  | Должность                    | Подпись | Дата       |
|----------------------|------------------------------|---------|------------|
| Петров Петр Петрович | Ведущий менеджер ООО "BoomQ" | Петров  | 09.09.2024 |
|                      |                              |         |            |
|                      |                              |         |            |

### 3 Список терминов и сокращений

| Термин | Полное наименование               |
|--------|-----------------------------------|
| НТ     | Нагрузочное тестирование          |
| БД, DB | База данных                       |
| UC     | User Case - Сценарий пользователя |
| TR     | Transaction - Транзакция          |
| ПО     | Программное обеспечение           |

## 4 Введение

### 4.1 Назначение документа

Методика НТ представляет собой подробное описание процессов и этапов проведения тестирования производительности платформы Boomq.

Документ преследует следующие цели:

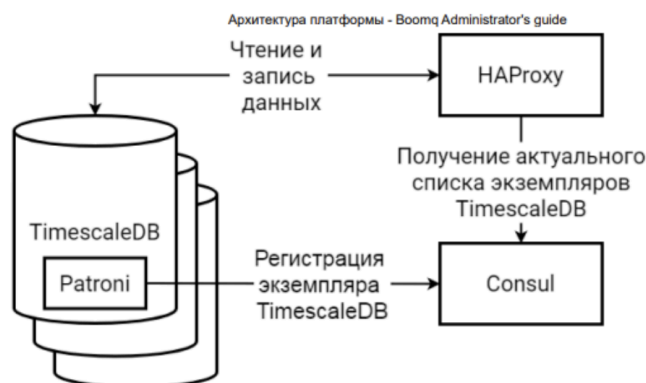
- описать стратегию тестирования производительности систем;
- описать планируемые этапы работ;
- описать объект исследования и конфигурацию тестового стенда;
- описать порядок передачи результатов проекта;
- описать рамки и ограничения тестирования;
- описать методики тестирования.

Методика НТ предназначена для специалистов бизнес-подразделений, менеджеров и технических специалистов Заказчика, а также будет использоваться при проектировании и проведении тестов специалистами «Перформанс Лаб».

### 4.2 Объект тестирования

Объектом тестирования является система Boomq, предназначенная для автоматизации нагрузочного тестирования информационных систем. Boomq позволяет командам эффективно работать над проектами, связанными с нагрузочным тестированием, через удобный веб-интерфейс. Система поддерживает распределенное тестирование с возможностью автоматического развертывания генераторов нагрузки и автоматической генерации отчетов.

Компоненты платформы :



#### 4.2.1 Frontend

Пользовательский интерфейс BoomQ.

#### 4.2.2 Telegraf on Docker

Агент для сбора метрик и данных в системе. Telegraf записывает собранные данные в TimescaleDB.

#### 4.2.3 Proxy NGINX

Прокси-сервер, который обеспечивает взаимодействие Frontend и Backend Boomq.

#### 4.2.4 HAproxy

Прокси-сервер, который обеспечивает взаимодействие платформы Boomq и TimescaleDB.

#### 4.2.4 Микросервисы Boomq

Обеспечивают функциональные возможности Boomq.

#### 4.2.5 MinIO

Файловое хранилище, в котором хранятся файлы, загружаемые пользователями, и файлы, которые создаются по результатам тестов, например, лог-файлы.

#### 4.2.6 Grafana

Инструмент для визуализации данных, полученных из TimescaleDB

#### 4.2.7 PostgreSQL

БД, которая содержит тестовые проекты, настройки и пользователей.

#### 4.2.8 TimescaleDB

БД которая содержит:

- системные метрики объекта тестирования;
- тестовые метрики, получаемые от JMeter;
- метрики нагрузочной станции, получаемые от Telegraf в контуре Boomq.

Экземпляров БД может быть несколько.

#### 4.2.9 Patroni

Приложение для управления репликаций между экземплярами TimescaleDB. Устанавливается на каждом экземпляре TimescaleDB.

Patroni регистрирует в Consul экземпляр TimescaleDB и получает информацию о других экземплярах БД. HAproxy получает актуальный список экземпляров TimescaleDB из Consul и данные о том, какие экземпляры поддерживает запись, а какие - чтение.



## 5 Цели и задачи

Целью данного нагрузочного тестирования является определение максимальной производительности системы Boomq и оценка её стабильности при повышенной нагрузке, а также выявление узких мест, которые могут ограничивать производительность системы в реальных условиях эксплуатации.

1. **Выявление максимальной производительности системы** — определение предела нагрузки, при которой система может стабильно функционировать, и оценка ее устойчивости при росте нагрузки до 200% от стандартной.
2. **Оценка стабильности системы при повышенной нагрузке** — анализ работы BOOMQ при экстремальных условиях для определения возможных отказов и снижения производительности.
3. **Определение узких мест в компонентах системы** — выявление компонентов системы, которые оказывают наибольшее влияние на снижение производительности при увеличении нагрузки.
4. **Проверка отказоустойчивости** — проверка работы системы при высоких нагрузках с учетом возможных сбоев и нарушений соединения, а также оценка способности системы к восстановлению и продолжению работы без значительного ухудшения производительности.
5. **Анализ влияния роста объема данных на производительность** — изучение влияния увеличения объема обрабатываемых данных на систему и её компоненты, включая базы данных и системы хранения.

### Основные задачи:

1. **Анализ профилей нагрузки** — изучение бизнес-процессов и операций в системе для определения критических операций, которые будут использованы для моделирования нагрузки.
2. **Эмуляция внешних систем и нагрузок** — настройка инструментов эмуляции внешних систем и операций, которые взаимодействуют с Boomq, для создания реальной нагрузки.
3. **Разработка сценариев нагрузочного тестирования** — создание тестовых сценариев, которые будут моделировать нагрузку до 200% от стандартной, включая генерацию запросов, данные, и симуляцию различных типов пользователей.
4. **Проведение нагрузочных испытаний** — проведение тестов в контролируемой среде с поэтапным увеличением нагрузки до 200% от обычных рабочих условий.
5. **Сбор и анализ данных** — сбор метрик производительности системы, включая использование процессора, памяти, скорости откликов и времени выполнения запросов.
6. **Поиск узких мест** — анализ результатов тестирования для выявления узких мест в системе и подготовка рекомендаций по оптимизации производительности.
7. **Подготовка отчетов** — создание детализированных отчетов с результатами тестирования, указанием на выявленные проблемы и предложениями по их устранению.

8. **Проведение повторного тестирования** — сравнение результатов после оптимизации с исходными показателями для оценки эффективности внедренных рекомендаций.

## 6 Ограничения тестирования

В рамках проводимого нагрузочного тестирования следует отметить следующие ограничения:

- Данное тестирование не является функциональным и не служит для выявления функциональных дефектов, в то же время, обнаруженные в ходе проведения работ дефекты регистрируются и передаются Заказчику.
- Тестирование не направлено на выявление дефектов в аппаратной части стенда.
- Не оценивается влияние загрузки каналов связи.
- Перед проведением тестирования на этапе создания нагрузочных скриптов версии компонент информационной системы фиксируются и не изменяются до окончания тестирования, за исключением случаев устранения ошибок, мешающих дальнейшему проведению работ по тестированию.
- Организация, работоспособность и доступность тестового стенда обеспечивается Заказчиком.



| Host                      | Система | Параметр  | Значение    |
|---------------------------|---------|-----------|-------------|
| Нагрузочная станция Boomq |         | CPU cores | 8           |
|                           |         | RAM       | 32GB        |
|                           |         | Hard      | 500 GB      |
|                           |         | Software  | OS Centos 7 |

Таблица 1 – Конфигурация серверов продуктивного стенда

## 8 Взаимодействие с внешними системами

**Эмуляция внешних систем:** Вместо реальной внешней системы для тестирования используется программная **заглушка**, которая эмулирует работу реальной системы. Заглушка реализована в виде эмулятора объекта тестирования и представляет собой скрипт на языке Java.

**Эмулятор объекта тестирования:** Эмулятор реализован для платформы Boomiq. Он получает нагрузочные данные от системы и имитирует поведение реального объекта тестирования.

**Генерация и передача сообщений:** Со стороны нагрузочной станции Boomiq в адрес объекта тестирования будут генерироваться JSON-сообщения. Сообщения отправляются на сервер для мониторинга и заглушек по адресу:  
<http://77.50.236.215:48997/stub>.

**Ответ заглушки:** Заглушка возвращает тот же JSON, который был получен от системы Boomiq. К каждому строковому элементу JSON-документа добавляется текстовый комментарий.

**Особенности работы:** Сообщения генерируются системой Boomiq во время процесса тестирования автоматически. Логика работы объекта тестирования специально не уточнялась, что дает возможность заглушке возвращать минимально необходимый ответ.

Таким образом, взаимодействие с внешними системами при нагрузочном тестировании эмулируется через заглушки и эмуляторы, обеспечивая масштабируемость и гибкость системы Boomiq при нагрузочных тестах.

[Что по поводу этой хрени?](#)

[Эмулятор "Браузер пользователя"](#)

[Взаимодействие с тестовой системой происходит путем HTTP запросов и ответов с отрисовкой графики в браузере. Для эмуляции будет сделан сервис, генерирующий запросы в соответствии со схемами работы пользователей и принимающий ответы.](#)

## 9 Стратегия тестирования

Основная цель — это определение максимальной производительности системы, подтверждение полученных данных и тестирование стабильности работы системы при длительных нагрузках.

### 9.1 Этап тестирования

Проект проведения нагрузочного тестирования делится на следующие этапы:

1. **Создание методики** (текущий документ).
2. **Подготовка среды тестирования** — обеспечение необходимой инфраструктуры и конфигураций для проведения нагрузочных испытаний.
3. **Разработка средств эмуляции нагрузки** — подготовка инструментов для моделирования реальных нагрузок на систему.
4. **Разработка сценариев тестирования** — написание скриптов, эмулирующих работу пользователей и процессов в системе.
5. **Разработка заглушек** — создание эмуляторов для внешних систем, с которыми взаимодействует система.
6. **Заполнение данных** — генерация и загрузка необходимых объемов тестовых данных в систему.
7. **Проведение тестирования:**
  - **Поиск максимальной производительности** — пошаговое увеличение нагрузки до тех пор, пока система не достигнет предела своей работоспособности (максимальной нагрузки, при которой еще выполняются критерии успешности).
  - **Подтверждение максимальной производительности** — повторное проведение тестов на уровне, близком к предельной нагрузке, для проверки устойчивости системы.
  - **Тестирование стабильности** — проверка устойчивости системы при длительном воздействии высокой (но не предельной) нагрузки, чтобы выявить возможные утечки ресурсов или деградацию производительности.

**Системный анализ** — анализ полученных данных для выявления узких мест и проблемных компонентов системы.

**Подготовка отчета** — сбор и представление результатов тестирования в отчете.

#### Критерии успешного завершения тестирования:

- Достигнута повторяемость результатов тестов на разных этапах.
- Все запланированные тесты выполнены корректно.
- Получены и зафиксированы метрики производительности (использование CPU, памяти, задержки и т.д.).
- Измерены и проанализированы времена отклика интерфейса.
- Тест стабильности считается успешным, если система стабильно функционирует под высокой нагрузкой в течение запланированного времени (например, 24 часа) без значительного ухудшения производительности.

Дополнительные критерии успешности могут быть уточнены в зависимости от конкретных требований заказчика.



## 10 Моделирование нагрузки

Для проведения нагрузочного тестирования будут разработаны специальные средства НТ, которые обеспечат адекватное моделирование реальных условий эксплуатации системы. В данном разделе описаны требования к средствам НТ и процесс моделирования нагрузки.

**Средства НТ** разрабатываются с использованием ПО LoadRunner v24.3, которое предназначено для создания и проведения нагрузочных тестов. Данное ПО позволяет эмулировать действия большого количества виртуальных пользователей (VU) и собирать метрики производительности.

**Эмулятор системы BoomQ** разрабатывается на платформе JDK 8 и запускается как Java-приложение. Эмулятор предназначен для имитации взаимодействия с внешними системами, что описано в разделе "8. Взаимодействие с внешними системами". Он обрабатывает запросы от тестируемой системы и генерирует ответы, необходимые для выполнения нагрузочных сценариев.

**Моделирование нагрузки от регламентных операций** осуществляется путем создания тестовых сценариев, собранных в единую группу и запускаемых одновременно. Интенсивность выполнения сценариев регулируется временем отклика системы и задержкой между итерациями.

### Интенсивность сценариев:

В процессе тестирования время выполнения каждого сценария и отклика системы фиксируется и считается постоянным.

Интенсивность выполнения сценария изменяется за счет регулирования количества виртуальных пользователей (VU) и величины задержек между их итерациями.

Если сумма времени отклика системы и выполнения сценария не превышает задержку между итерациями, увеличение числа VU приведет к пропорциональному увеличению суммарной нагрузки.

### Изменение нагрузки:

Моделирование изменения нагрузки производится путем изменения количества VU, одновременно выполняющих сценарии, и величины задержек между итерациями.

Величина задержки и количество VU для различных сценариев рассчитываются на основе данных, предоставленных из баз данных и статистики реальной системы. Эти расчеты проводятся на этапе подготовки тестовой среды и скриптов НТ.

### Поддержка тестовой среды:

Для оптимизации и точного моделирования нагрузок используется Excel-шаблон, в который загружаются данные из системы. На основе этих данных рассчитываются необходимые параметры для каждого сценария — время выполнения, задержки между итерациями и количество виртуальных пользователей.

Таким образом, разработанные средства НТ позволят гибко управлять интенсивностью нагрузки и моделировать реальные условия эксплуатации системы, что обеспечит точность и надежность проводимых нагрузочных испытаний.

# 11 Тестовый стенд

## 11.1 Архитектура тестового стенда

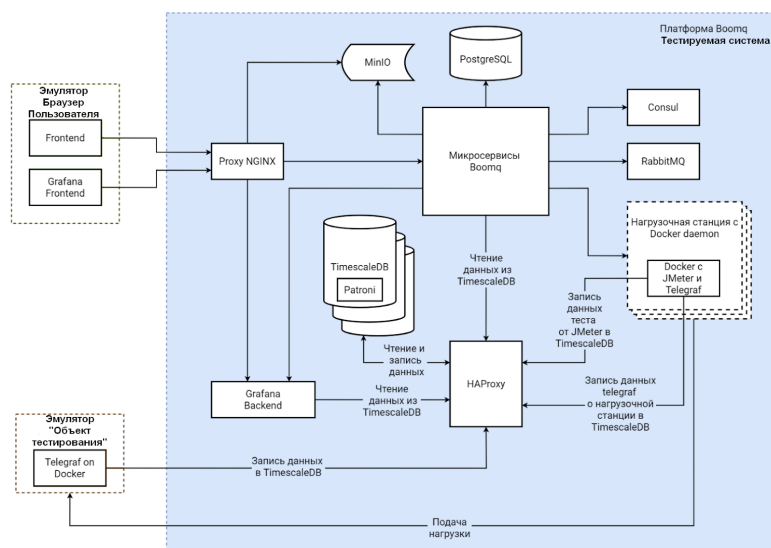


Рисунок 3 – Архитектурная схема тестового стенда

## 11.2 Конфигурация тестового стенда:

Таблица 3 – Конфигурация тестового стенда

| Тип/имя сервера                            | Конфигурация   | Кол-во в прод. | Кол-во для нагрузочного тестирования | ОС                 |
|--|--|----------------|--------------------------------------|--------------------|
| Тестовый стенд<br>IP: 77.50.236.214:2032   | 8x Intel(R) Xeon(R) CPU E5-2690 v2 @3.00GHz, 16GB, 100GB | -              | 1                                    | CentOS 7           |
| Сервер заглушек<br>IP: 77.50.236.215:20022 | 8x Intel(R) Xeon(R) Gold 6326 CPU @2.90GHz, 32GB, 200GB  | -              | 1                                    | Ubuntu 22.04.3 LTS |
| Сервер с нагрузочными станциями Boomq      | 8x CPU, 32GB, 500GB                                      | -              | 1                                    | CentOS 7           |

## 11.3 Конфигурация ПО

(Спросить какие версии программы)

Для проведения нагрузочного тестирования на тестовом стенде используются следующие программные компоненты:

1. **Операционная система:**
  - На серверах используется ОС **CentOS 7** для всех компонентов тестовой среды.
  - Серверы заглушек используют ОС **Ubuntu 22.04.3 LTS** для поддержки эмуляции внешних систем.
2. **Платформа для тестирования:**
  - **LoadRunner v24.3**
3. **Средства мониторинга:**
  - **Telegraf**
  - **Grafana**
4. **Базы данных:**
  - **PostgreSQL**
  - **TimescaleDB**
5. **Инструменты для эмуляции внешних систем:**
  - **Java-приложений (JDK 8)**
6. **Контейнеризация и виртуализация:**
  - **Docker**

## 11.4 Тестовые данные для средств НТ

| Название операции | Новая Интенсивность оп/ч | Время выполнения (сек) |
|-------------------|--------------------------|------------------------|
| Регистрация       | 195                      | 10                     |
| Создание теста    | 1170                     | 8                      |
| Запуск теста      | 351                      | 8                      |
| Создание отчета   | 780                      | 8                      |
| Создание тренда   | 1170                     | 8                      |

**Длина теста:** для тестирования платформы BoomQ тесты должны длиться не менее 1 часа.

**Количество VU:** количество пользователей выбирается случайным образом в диапазоне от 1 до 5, что позволяет эмулировать реальное использование системы в условиях переменной нагрузки.

## 11.5 Методика удаления тестовых данных после тестирования

Для обеспечения чистоты тестовой среды после завершения тестирования применяется следующая методика удаления тестовых данных:

**Точка восстановления:** Перед началом тестирования создается точка восстановления тестового стенда, которая сохраняет исходное состояние базы данных и системы.

**Откат данных:** После завершения тестирования среда восстанавливается до момента, когда база данных содержит исходные тестовые данные, включая:

- 25 000 зарегистрированных пользователей.
- 25 000 созданных тестов.
- 25 000 созданных отчетов и трендов.

**Автоматическое удаление данных:** Все данные, сгенерированные в ходе тестирования, удаляются автоматически через систему отката до точки восстановления. Это гарантирует, что тестовая среда возвращается в первоначальное состояние, исключая возможность накопления данных, которые могут повлиять на последующие тесты.

### 11.6 Прочие требования, предъявляемые к тестовой среде

Аккаунты, выданные команде нагрузочных тестировщиков, не должны иметь никаких лицензионных ограничений, иначе тестирование будет проведено в том виде и объеме, какой будет наложен имеющимися лицензионными ограничениями.

## 12 Анализ статистики (так и не понял откуда и что мы должны вписать)

### 12.1 Наиболее часто используемые операции

На основании проведенного анализа бизнес процессов и анализа предоставленной Заказчиком статистики выполнения операций в промышленной системе за неделю 18-24.11.2013, был выбран день с максимальной интенсивностью и час пиковой нагрузки (ЧПН) с 12:00 до 13:00 и определены наиболее часто используемые и ресурсоемкие операции.

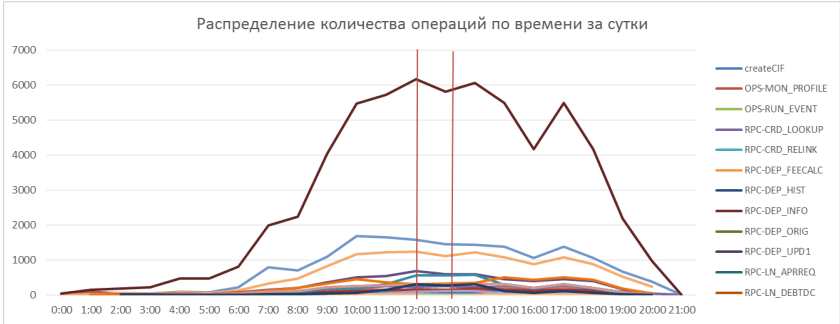


Рисунок 4 – График распределения операций по времени за сутки

В таблице ниже представлен список наиболее часто используемых операций.

Таблица 4 – Список операций с разбивкой по типам по типам в ЧПН

| № | Названия операций | Количество | % от общего количества | % с нарастающим итогом |
|---|-------------------|------------|------------------------|------------------------|
| 1 | RPC-LN_DELREQF    | 1578       | 26%                    | 26%                    |
| 2 | RPC-LN_INFO       | 1232       | 20%                    | 46%                    |
| 3 | RPC-LN_LOOKUP     | 685        | 11%                    | 57%                    |
| 4 | RPC-LN_ORIG       | 560        | 9%                     | 66%                    |

| №  | Названия операций | Количество | % от общего количества | % с нарастающим<br>итогом |
|----|-------------------|------------|------------------------|---------------------------|
| 5  | updateCIF         | 320        | 5%                     | 71%                       |
| 6  | RPC-LN_PAYLK      | 308        | 5%                     | 76%                       |
| 7  | RPC-LN_PMTSCH     | 307        | 5%                     | 81%                       |
| 8  | RPC-LN_APPREQ     | 286        | 5%                     | 86%                       |
| 9  | RPC-CRD_LOOKUP    | 239        | 4%                     | 90%                       |
| 10 | RPC-LN_INSADD     | 170        | 3%                     | 93%                       |
| 11 | createCIF         | 143        | 2%                     | 95%                       |
| 12 | RPC-LN_DISB       | 115        | 2%                     | 97%                       |
| 13 | RPC-LN_UPD1       | 101        | 2%                     | 99%                       |
| 14 | RPC-LN_INS0       | 40         | 1%                     | 99%                       |
| 15 | RPC-LN_HIST       | 33         | 1%                     | 100%                      |
| 16 | OPS-MON_PROFILE   | 21         | 0%                     | 100%                      |



| №  | Названия операций | Количество | % от общего количества | % с нарастающим итогом |
|----|-------------------|------------|------------------------|------------------------|
| 17 | RPC-LN_PAYLKD     | 20         | 0%                     | 100%                   |
| 18 | RPC-DEP_INFO      | 5          | 0%                     | 100%                   |
| 19 | RPC-LN_EDC        | 5          | 0%                     | 100%                   |
| 20 | RPC-ZCIFPKG_LIST  | 3          | 0%                     | 100%                   |
| 21 | RPC-LN_FER        | 2          | 0%                     | 100%                   |
| 22 | RPC-LN_TRN_PROC   | 2          | 0%                     | 100%                   |
|    | Общий итог        | 6175       | 100%                   |                        |

Зеленым отмечены операции, выбранные для профиля нагрузки «Операционная деятельность».

В этом разделе проводится анализ данных, собранных в ходе тестирования. Анализ включает следующие аспекты:

**1. Наиболее часто используемые операции:**

- Определение операций, которые чаще всего выполнялись в ходе тестирования.
- Определение операций, которые являются наиболее ресурсоемкими.

**2. Метрики производительности:**

- Время отклика системы на различные операции.
- Загрузка процессора (CPU) и использование памяти.
- Утилизация дисковой подсистемы (время отклика дисков, операции чтения/записи).
- Пропускная способность сети.

**3. Распределение нагрузки:**

- Определение пиковых нагрузок и их влияние на производительность системы.
  - Анализ работы системы в условиях высокой нагрузки.
4. **Ошибки и отклонения:**
- Количество ошибок, возникших во время выполнения тестов
  - Время восстановления системы после ошибок или перегрузок
  - Сравнение фактического поведения системы с ожидаемым.
5. **Заключение по результатам анализа:**
- Определение узких мест в системе, которые могут снизить ее производительность.
  - Рекомендации по оптимизации и улучшению производительности.

### Как это оформить

1. **Собрать статистику** после тестов: это метрики (время отклика, загрузка CPU, количество операций и т.д.), которые ты соберешь с помощью инструментов мониторинга, таких как Grafana, Telegraf и др.
2. **Графическое представление:** если ты используешь инструменты вроде Grafana, то метрики можно представить в виде графиков и диаграмм, которые помогут проанализировать результаты.
3. **Выводы:** после анализа нужно сделать выводы о том, как система справляется с нагрузкой, есть ли узкие места, и какие оптимизации могут быть рекомендованы.

Если ты пока не собрал данных, то начни с планирования того, какие метрики ты будешь отслеживать во время тестирования.

## 13 Профили нагрузки

### 13.1 Процессинговая деятельность BoomQ

Таблица 5 – Список операций

| № | Название операции | Операций в час | Распределение % |
|---|-------------------|----------------|-----------------|
| 1 | Регистрация       | 195            | 5.31%           |
| 2 | Создание теста    | 1170           | 31.91%          |
| 3 | Запуск теста      | 351            | 9.75%           |
| 4 | Создание отчета   | 780            | 21.27%          |
| 5 | Создание тренда   | 1170           | 31.91%          |
|   | Итого             | 3666           | 100%            |

## 14 Сценарии пользования

Таблица 6 – Перечень эмулируемых операций

| ID<br>ТЕСТА | НАЗВАНИЕ<br>КОМПОНЕНТА | НАЗВАНИЕ СЦЕНАРИЯ | НАЗВАНИЕ СКРИПТА   |
|-------------|------------------------|-------------------|--|
| UC01        | Регистрация            | Register          | UC01_Register_TR01_mainPage<br>UC01_Register_TR02_login_as_admin<br>UC01_Register_TR03_nagruzka<br>UC01_Register_TR04_edit<br>UC01_Register_TR05_add_new_team_member<br>UC01_Register_TR06_copy_link<br>UC01_Register_TR07_athorize_new_user<br>UC01_Register_TR08_create_password |
| UC02        | Создание теста         | Create Test       | UC02_CreateTest_TR01_mainPage<br>UC02_CreateTest_TR02_login<br>UC02_CreateTest_TR03_add_new_test<br>UC02_CreateTest_TR04_save_test   |
| UC03        | Запуск теста           | Test Launch       | UC_03_TestLaunch_TR01_mainPage<br>UC_03_TestLaunch_TR02_login<br>UC_03_TestLaunch_TR03_tests<br>UC_03_TestLaunch_TR04_launch<br>UC_03_TestLaunch_TR05_run_test   |
| UC04        | Создание отчета        | Create Report     | UC04_CreateReport_TR01_mainPage<br>UC04_CreateReport_TR02_login<br>UC04_CreateReport_TR03_reports<br>UC04_CreateReport_TR04_add_new_report<br>UC04_CreateReport_TR05_add_items<br>UC04_CreateReport_TR06_back_to_reports   |

| ID<br>ТЕСТА | НАЗВАНИЕ<br>КОМПОНЕНТА | НАЗВАНИЕ СЦЕНАРИЯ | НАЗВАНИЕ СКРИПТА  |
|-------------|------------------------|-------------------|---|
| UC05        | Создание тренда        | Create Trend      | UC05_CreateTrend_TR01_mainPage<br>UC05_CreateTrend_TR02_login<br>UC05_CreateTrend_TR03_trends<br>UC05_CreateTrend_TR04_select_test<br>UC05_CreateTrend_TR05_add_charts_tables<br>UC05_CreateTrend_TR06_save |

Примечание. При эмуляции, каждая операция будет в свою очередь разбита на транзакции (логин, открытие формы, и др.).

15 Наполнение БД

**Создание пользователей:** Все учетные записи создаются автоматически с помощью скриптов, имитирующих реальные регистрационные процессы. Это гарантирует, что пользователи имеют уникальные данные, необходимые для тестирования.

**Создание тестов:** Каждый зарегистрированный пользователь создает 1-2 теста, имитируя стандартный рабочий процесс на платформе.

**Генерация отчетов и трендов:** По результатам выполненных тестов от имени каждого пользователя создаются отчеты и тренды, что позволяет моделировать реальную активность пользователей платформы Boomq.

**Проверка данных:** После завершения процесса наполнения БД проводится проверка корректности созданных данных, чтобы убедиться, что все учетные записи, тесты, отчеты и тренды соответствуют требованиям тестирования.

Эти данные обеспечат достаточно реалистичную и репрезентативную нагрузку для того, чтобы протестировать ключевые функции системы и определить её производительность и стабильность под высокой нагрузкой

| Наполнение БД Profile                         | 2024г. |
|---|--------|
| Общее кол-во зарегистрированных пользователей | 25 000 |
| Общее кол-во созданных тестов                 | 25 000 |
| Общее кол-во созданных отчетов                | 25 000 |
| Общее кол-во созданных трендов                | 25 000 |

## 16 Планируемые тесты

### 16.1 Перечень типов тестов

В данном разделе приводится список типов тестов, которые планируется провести для оценки производительности системы.

Таблица 9 – Перечень планируемых тестов

| № | Тест   | Описание  |
|---|--|---|
| 1 | Тест максимальной производительности системы       | Целью теста является определить максимальную нагрузку, при которой система начинает снижать производительность (увеличение времени отклика или ошибки).       |
| 2 | Тест подтверждения максимальной производительности | Повторный тест на уровне, близко к максимальной нагрузке( например 90% от L0), для подтверждения данных о снижении производительности при нагрузке.           |
| 3 | Тест стабильности системы                          | Тестирование системы при длительной нагрузке (80% от максимальной) для проверки стабильности работы и выявления возможных проблем, таких как утечки ресурсов. |

### 16.2 Планируемые тесты

В данном разделе приводится описание каждого теста, включая его продолжительность, количество запусков и описание процесса.

Таблица 10 – Перечень планируемых тестов

| № | Тест   | Длительность теста | Кол-во запусков | Описание теста  |
|---|--|--------------------|-----------------|---|
| 1 | Тест максимальной производительности системы | 1 час 40 мин.      | >=1             | При тестировании происходит пошаговое увеличение нагрузки с нуля до предельной (с шагом 20% от плановой). Пошаговое увеличение происходит до тех пор, пока не нарушится один из критериев успешности. Время работы теста на каждом шаге (ступени) после стабилизации нагрузки (этап стабилизации нагрузки равен 5 минутам) составляет 10 мин. По результатам устанавливается уровень нагрузки L0 (последняя ступень нагрузки, на которой не были нарушены критерии успешности). |

| № | Тест   | Длительность теста   | Кол-во запусков | Описание теста  |
|---|--|--|-----------------|---|
| 2 | Тест подтверждения максимальной производительности | зависит от результатов "Теста максимальной производительности" | <b>&gt;=1</b>   | Тест проводится на ступени нагрузки, предшествующей L0 (или на уровне нагрузки 90% от L0). Длительность стабильной нагрузки не менее 1 часа. Если в процессе тестирования система оказалась недогружена или перегружена, то значение нагрузки корректируется и второй тест проводится повторно. В случае увеличения нагрузки новый уровень может быть рассчитан на основе данных об утилизации ресурсов. Результатом тестирования является максимальный достигнутый уровень нагрузки (обозначается Lmax). |
| 3 | Тест стабильности системы                          | 8 ч.   | <b>1</b>        | Тест проводится на уровне нагрузки Lstab = 80% от Lmax. Длительность стабильной нагрузки не менее 24 часов. В ходе теста фиксируются все отклонения от "нормального" поведения системы, в т.ч. деградация производительности, утечки.   |

## 16.3 Критерии успешности проведенного теста

### 1. Тест максимальной производительности системы

- Тест считается успешным, если система достигает предельной нагрузки, при которой наблюдаются критические задержки в отклике или ошибки, и это состояние подтверждается при повторном тестировании.
- Если при повторном тестировании система демонстрирует те же результаты (начало деградации производительности на том же уровне нагрузки), то это означает, что был найден максимальный уровень производительности системы.

### 2. Тест стабильности системы

- Тест стабильности считается успешным, если система способна поддерживать стабильную работу при нагрузке на уровне 80% от максимальной нагрузки, установленной в тесте максимальной производительности, в течение 24 часов.
- В процессе теста не должно возникать критических ошибок, деградации производительности или утечек ресурсов.

### 3. Общие критерии успешности для всех тестов

- В ходе тестирования запросы выполняются с частотой, соответствующей заранее установленным профилям нагрузки.
- Допускается не более 5% ошибок на один сценарий, и не более 10% ошибок на всех сценариях суммарно.
- Время отклика системы должно находиться в пределах, допустимых для рабочих условий эксплуатации системы.



## 17 Требования к производительности

Таблица 11 – Требования к времени отклика по типам операций

| Операция                  | Требование к времени отклика (сек) |
|---------------------------|------------------------------------|
| Операционная деятельность |                                    |
| Регистрация               | 10                                 |
| Создание теста            | 8                                  |
| Запуск теста              | 8                                  |
| Создание отчета           | 8                                  |
| Создание тренда           | 8                                  |

Эти требования к времени отклика определяют, как быстро система должна выполнять основные операции в условиях рабочей нагрузки.

### Дополнительные требования к производительности:

#### Интенсивность обработки операций:

- Система должна быть способна обрабатывать операции с интенсивностью, указанной в профиле нагрузки (см. раздел "Моделирование нагрузки").

#### Утилизация процессорных мощностей:

- Утилизация процессорных мощностей не должна превышать 80%. Это значение может корректироваться в зависимости от реальных данных, полученных в ходе тестов, но в любом случае рекомендуется поддерживать его на уровне, который не приводит к деградации производительности.

#### Утилизация дисковой подсистемы:

- Средняя утилизация дисковой подсистемы не должна превышать 90%.